REMARKS

The final Office Action mailed March 10, 2004, considered claims 1-25. Claims 1-16 and 18-24 were rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. Claims 1-6 and 9-25 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Bredenberg (U.S. Patent No. 5,826,253) in view of Beizer (U.S. Patent No. 6,240,414)[1].

By this paper, claims 1, 18 and 19 have been amended, such that claims 1-25 remain pending for consideration, of which claims 1, 17, 19, 25 are the independent claims at issue. The amendments to the claims have been made to overcome the rejections made under 35 U.S.C. § 112. In particular, the objected to language regarding the implementation of a command even when the command "is not in conflict with other high-level document commands," has been deleted. This element, however, is already inferred by the claims and the disclosure of the specification. Accordingly, by deleting this element, Applicants do not acquiesce that the scope of the invention does not cover this embodiment. To the contrary, the specification clearly describes the implementation of high-level document commands that are not in conflict. In particular, the specification fails to recite even a single embodiment in which a conflict would prevent the implementation of a high-level document command. Accordingly, for at least this reason, Applicants respectfully submit that the specification does support such embodiments. Nevertheless, this element has been deleted to expedite prosecution and to bring more attention to other distinctions that are more clearly at issue, e.g., the act of notifying client applications of a high-level document command prior to its implementation, as described below.

As reflected above, in the listing of claims, claim 1 is directed to a method for altering a high-level document command including, receiving a single high-level document command meeting certain criteria, the single high-level document command including one or more operations, *prior to implementing the single high-level document command*, identifying one or more client applications that are to be notified of the receipt of only high-level document commands meeting certain criteria, also *prior to implementing the single high-level document*

---

[1] Although the prior art status of the cited art is not being challenged at this time, Applicants reserve the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

command, an act of notifying the one or more identified client applications that the single high-level document command meeting the certain criteria has been received, notification of the one or more client applications being triggered solely as a result of receiving the single high-level document command, receiving modifying instructions from the one or more client applications on how to affect the implementation of the single high-level document command, and altering the one or more operations included in the single high-level document command according to the modifying instructions.

As described throughout the specification, notifying client applications of the receipt of a high-level document command, *PRIOR TO IMPLEMENTATION*, is useful "so that the client application may return back implementation instructions on how to implement the high-level document command. ...For example, the instruction may be for preventing the implementation of the high-level document command altogether. The instruction may also be to change how the high-level document command is implemented. The instruction may even be for implementing one or more high-level document commands in addition to the received high-level document command. Thus, the notified third party client application is given extensive control over how the high-level document command is implemented..." Page 5, line 16 thru Page 6, line 2.

This is to be contrasted with the prior art systems wherein "the notification is only dispatched after the high-level document command is implemented. There is nothing the client application can do to affect how the high-level document command is implemented..." Page 4, line 11-13.

The art cited by the Examiner represents the prior art systems recognized by the pending application. In particular, the cited disclosure in Beizer relates to resolving conflicts that arise as a result of implementing high-level document commands. These conflicts can be detected, for example, when documents are saved or later reconciled. (Col. 5, ll. 16-17) In one example, it is "the last party to save their WorkFolder [that] will be notified of the conflict when the WorkFolder is saved..." Col. 6, ll. 64-65.

Accordingly, the cited disclosure in Beizer relates to resolving conflicts that are detected in saved documents. This, however, is to be contrasted with notifying a client application prior to implementing a save or copy command on a document, as enabled by the practice of the present application. (note: a high-level document command is a command corresponding to an "operation to be performed on a document" Page 13, ll. 19-20) Accordingly, notifying a party

that a conflict exists between saved documents, after the documents have been saved, does not meet the burden of notifying a client application of a high-level document command Prior to Implementing the High-Level document command.

Additionally, with regard to the Examiner's argument that Beizer teaches the act of notifying the client applications, Applicants would also like to respectfully point out that the disclosure cited by the examiner relates to notifying humans, who are the users of the computing systems, of a conflict. In fact, the Examiner rejects the claim in part by stating how Beizer "notifies the correct parties," citing to col. 5, ll. 30-40; col. 7, ll. 1-10. This, however is not what is recited in the claims regarding the notification of client *applications.*[2]

Beizer further clarifies this distinction in the passage cited by the Examiner in Col. 7, making it clear that it is the user that is notified to override or otherwise resolve a conflict. Accordingly, Applicants respectfully submit that the acts of notifying client applications prior to implementing a high-level document command is not met by notifying a user of a conflict that resulted from having already implemented a high-level document command, as suggested by the Examiner.

Applicants also submit that these elements, which are present in all of the pending claims, are also not taught or suggested by Bredenberg. To the contrary, Bredenberg teaches that a client user is only notified of changes that are already made to the records of interest to the client. In particular, Bredenberg teaches that the client indicates "the records which the client desires to know if any changes are posted." Col. 5, ll. 10-11. Bredenberg merely notifies the client when an update has already occurred, so as to notify the client when records of interest in a local cache might be stale. (Abstract; Col. 5, ll. 35-36). Accordingly, Bredenberg's notifying clients after an update has already occurred should be distinguished from the pending claims in which client applications are notified of a high-level document command prior to implementing the high-level document command.

Accordingly, with regard to the combined teachings of Beizer and Bredenberg, notifications are only sent when two conditions are met: 1) receiving an update (implementing

---

[2] Analogizing humans with computing applications was also made in previous communications with the Examiner in which the client applications were considered by the Examiner to be analogous with a security officer. Applicant respectfully submits, however, that considering a human to be analogous with a computing application does not provide a reasonable interpretation of the claim elements. Inasmuch as this is the second time this has happened, Applicant respectfully requests that the Examiner apply a more reasonable reading of the claim elements in the future.

Page 12 of 14

the high-level document command) and 2) detecting that the implemented high-level document command, e.g., update, conflicts with another update. This, however, is contrasted with the pending claims that require the client applications to be notified of the high-level document command prior to implementation so that they can control the implementation of the high-level document command.

Accordingly, for at least the foregoing reasons, Applicant respectfully submits that Beizer and Bredenberg fail, either singly or in combination, to teach or suggest the methods recited in the pending claims.

Although the foregoing remarks have been primarily directed to claim 1, they also apply to the other independent claims (17, 19 and 25) at issue, and therefore, to all of the corresponding dependent claims. In particular, with regard to claim 17, the cited art fails to teach or suggest "receiving a single high-level document command meeting certain criteria" and "notifying the one or more identified client applications that a high-level document command meeting the certain criteria has been received *prior to implementing* the single high-level document command and solely in response to receiving the single high-level document command," (emphasis added) as discussed above with regard to claim 1, particularly when viewed in combination with the other recited limitations of claim 17.

Claim 19 also includes the same elements recited above in claim 1, inasmuch as claim 19 is directed to a computer-program product for implementing the method recited in claim 1. Thus, the cited art fails to teach or suggest the limitations of claim 19 at least for the reasons provided with respect to claim 1.

With regard to claim 25, the cited art fails to teach or suggest "a notification component that is configured to send a notification . . . triggered solely as a result of receiving the single high-level document command" "and *prior to implementation* of any operations included in single high-level document command" and "a database engine configured to alter one or more operations included in the single high-level document according to received instructions when implementing the single high-level document command," for at least the reasons cited above with regard to claim 1 and when viewed in combination with the other limitations of claim 25.

For at least the forgoing reasons, Applicant respectfully submits that the pending claims, 1-25 are now in condition for allowance. In the event that the Examiner finds remaining
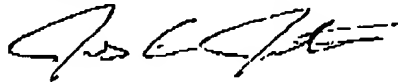
impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this __9__ day of July 2004.

Respectfully submitted,

RICK D. NYDEGGER
Registration No. 28,651
JENS C. JENKINS
Registration No. 44,803
Attorneys for Applicant

Customer No. 022913

RDN:JCJ:cm
W:\13768\173\CM0000003948V001.DOC

Page 14 of 14

PAGE 23/23 * RCVD AT 7/9/2004 9:22:44 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:18013281707 * DURATION (mm-ss):06-40